

# PIXEL 3D

PROFESSIONAL





AXIOM SOFTWARE

DEMO COPY  
NOT FOR RESALE

***PIXEL 3D***  
***PROFESSIONAL***

USER'S  
MANUAL

DEMO COPY  
NOT FOR RESALE

Axiom Software  
541 West 98th Street # 322  
Bloomington, MN 55420  
(612) 888-0352  
FAX (612) 888-0537

Copyright 1990, 1991, 1992 by Axiom Software. All rights reserved worldwide.

Pixel 3D Professional is a trademark of Axiom Software. Under the copyright laws, neither this manual nor the software described in this manual may be copied, in whole or in part, without the written consent of Axiom Software.

LightWave 3D is a trademark of NewTek, Inc. Amiga is a trademark of Commodore-Amiga, Inc. Imagine and Turbo Silver are trademarks of Impulse Corporation, Sculpt 3D is a trademark of Byte by Byte Corporation, 3D Professional is a trademark of Progressive Peripherals. VideoScape 3D is a trademark of Aegis Corporation, AutoCAD is a trademark of Autodesk Corporation, Art Department Professional is a trademark of ASDG, Inc and Deluxe Paint is a trademark of Electronic Arts, Inc. Any other products mentioned are trademarks of their respective companies.

### **Disclaimer of Warranty**

Axiom Software makes no warranties, either expressed or implied, as to the function, application or suitability of this product. The manual and accompanying software is sold as is, and Axiom Software will in no event be liable for direct, indirect, incidental or consequential damages resulting from any defect, error or failure for any reason.

### **License Agreement**

You have the right to use this software on ONE computer under ONE operating system. Duplication of this software for any purpose other than backup protection is a direct violation of this license agreement.

Printed in the U.S.A

## Table of Contents

1.0	Introduction	1
1.1	What is PixPro?	1
1.2	Hardware requirements	1
1.3	Installation	1
2.0	PixPro Basics	3
2.1	Navigating the Screen	3
3.0	Loading and Saving Objects	5
3.1	Loading Objects	5
3.2	Saving Objects	6
4.0	Loading Text and Bitmap Images	7
4.1	Loading Text	7
4.2	Loading Bitmap Images	8
5.0	Undo	9
6.0	The View Window	10
6.1	View Mode Buttons	10
6.2	Mouse Controls	11
6.3	Control Buttons	11
6.4	Coordinates	12
6.5	Display Mode Buttons	13
7.0	Object Editing Functions	15
7.1	Edit	15
7.2	Pick	15

7.3	Unpick	16
7.4	UnAll	16
7.5	Delete	16
7.6	Subdivide	16
7.7	Linked	16
7.8	Swap	16
8.0	Object Manipulation Functions	18
8.1	Smooth	18
8.2	Extrude	21
8.3	Bevel	24
8.4	Spin	26
9.0	Data Settings	28
9.1	Polygon Flipping	28
9.2	Polygon Division	29
9.3	Polygon Doubling	29
9.4	Polygon Reduction	30
9.5	Vertex Reduction	31
9.6	Polygon Color	31
9.7	Selected Data	32
10.0	Configuring Bitmap load Settings	33
10.1	Color	33
10.2	Monochrome	33
10.3	Grid	33
10.4	Smooth/Extrude/Bevel/Spin Settings	34
11.0	Setting Defaults	36
11.1	System Memory	36
11.2	Text Color	36
11.3	Save Default	37
11.4	Load and Save Configurations	37

12.0 Help	39
13.0 Quit	40
14.0 Appendices	41
14.1 Keyboard Shortcuts	41
14.2 Tips and Tricks	43
14.3 Notes On Specific Object Formats	47
14.3.0 Lightwave 3D	47
14.3.1 3D Pro	47
14.3.2 Turbo Silver/Imagine	47
14.3.3 AutoCAD DXF	48
14.3.4 Wavefront	48
14.3.5 Digital Arts	48
14.3.6 Videoscape BINary/Videoscape ASCii	48
14.3.7 Sculpt 3D	48
14.3.8 Draw 4D/ Draw 4D Pro	49
14.3.9 Caligari	49
14.4.0 Router Bit	49
14.4.1 Vista Pro	49
14.4.2 Scenery Animator	49
15.0 Tutorials	50
15.1 Converting text into an object	50
15.2 Converting a bitmap into an object	52
16.0 Technical Support and Q&A	55



# **1.0 Introduction**

## **1.1 What is PixPro?**

Pixel 3D Professional (PixPro) is a multi-function 3D object utility program for graphics professionals. PixPro was developed to simplify the task of creating and converting 3D objects.

PixPro's abilities include autotracing functions used to convert IFF bitmaps and entered bitmap fonts into 3D objects. Once the object is created, PixPro allows you to easily edit individual parts or whole objects in a variety of ways.

PixPro contains 3D object load and save functions that allow 3D objects to be easily converted from one file format to another. PixPro supports 14 major file formats for loading and saving objects.

Additional capabilities include data manipulation functions, allowing for maximum object efficiency, and a complete on-line help facility.

## **1.2 Hardware Requirements**

PixPro requires a minimum of 1 megabyte of chip ram. More ram will allow you to create more complex objects.

## **1.3 Installation**

PixPro is not copy protected and can be easily installed on a hard drive. Make a backup copy of the PixPro disk and store the original in a safe place. To install PixPro on a



hard disk, simply use the Workbench to drag the PixPro drawer icon from the PixPro disk into the appropriate drawer on the hard disk. Users of floppy disk systems may run the program directly from the PixPro disk. There are no special libraries required.

## 2.0 PixPro Basics

### 2.1 Navigating the Screen

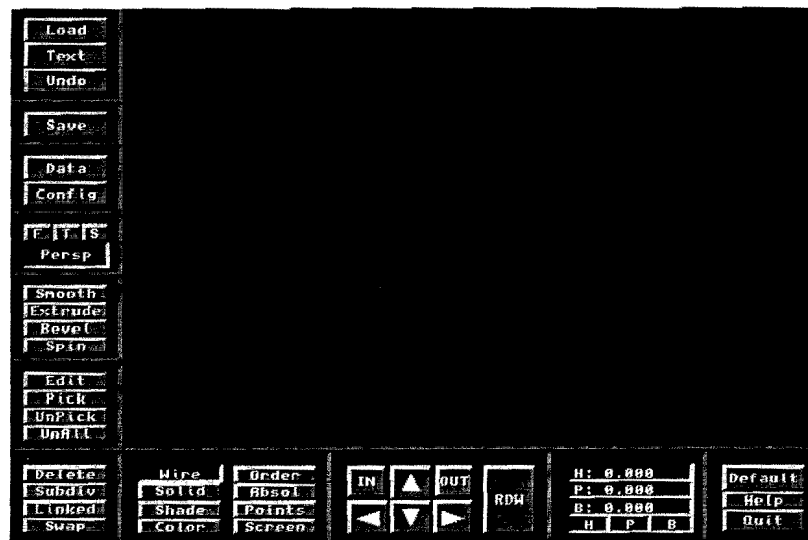
All of PixPro's functions are accessed via screen buttons and keyboard equivalents. There are no pull down menus. The screen is divided into logical sections.

The largest area of the screen is the view window where your object is displayed.

In the top left corner are buttons for **Load**, **Save**, **Text** and **Undo**. These are used for handling objects and bitmaps.

Below the **Save** button are **Data** and **Bitmap Configuration** buttons.

The middle left of the screen contains view mode buttons for seeing your object from the either the **front**, **top**, **side** or a **perspective** view.



The middle to lower left of the screen contains object manipulation and editing buttons.

Along the bottom of the screen are arranged the display buttons as well as the movement control panels.

Finally, in the lower right hand corner, you'll find the **Default Settings** button, the online-Help function and the **Quit** button.

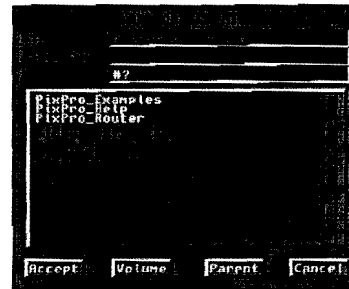
## 3.0 Loading and Saving Objects

### 3.1 Loading Objects (Keyboard Equivalent: l)

PixPro will automatically recognize and load any of the following fifteen 3D object types:

LightWave 3D  
Imagine  
3D Professional  
Sculpt 3D  
Turbo Silver  
VideoScape ASCII  
DXF AutoCAD  
VideoScape Binary  
Wavefront  
Draw 4D  
Digital Arts  
Draw 4D Professional  
Caligari  
Scenery Animator  
Vista Pro

When you click on the **Load** button, the **Load Object File** requester appears. You can load an object either by selecting or typing in the path and filename and then clicking **Accept** or simply by double-clicking on the filename. You can load only one object into memory at a time. If there is an object currently in memory, a requester will appear giving you the opportunity



to either **Continue** or **Cancel**. If the selected file is not found or is not a supported format, an error message is displayed and the loading process is aborted.

### 3.2 Saving Objects (Keyboard Equivalent: s)

Clicking on the **Save** button brings up the **Output Format** requester, allowing you to choose the object format you wish to save in. The following formats are supported:

LightWave 3D

Imagine

3D Professional

Sculpt 3D

Turbo Silver

VideoScape ASCII

DXF AutoCAD

VideoScape Binary

Wavefront

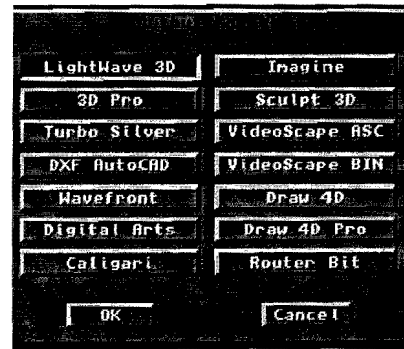
Draw 4D

Digital Arts

Draw 4D Professional

Caligari

Pixel Pro Router Bit



The output format defaults to that of the last object loaded. Clicking on **OK** calls up the **Save Object File** requester, where you can specify the directory and the name of the object to be saved. A file extension appropriate to the file format is automatically appended to the original filename. If the object filename already exists, you will be prompted to either save the new data over the old or cancel the operation.

## 4.0 Loading Text and Bitmap Images

One of the most powerful features of PixPro is it's ability to convert bitmap text or pictures to 3D objects. After loading a bitmap, some of the capabilities include standard extrusion, smoothing, beveling and spinning on an axis.

### 4.1 Loading Text (Keyboard Equivalent: t)

When you click on the **Text** button, the **Enter Text to Convert** requester is displayed. If an object is already in memory, you can remove it or cancel the text command. Type in the text you want, and press Return or click on **OK**. Click **Cancel** if you decide not to convert the text.



Next you will see the **Font Name/Size** requester, where you can choose the bitmap font and size of your text. The text function works best with large font sizes, preferably 100 points or larger. Double-click on the preferred font directory, then choose the font size and click on the **Accept** button. If the selected font is not found, an error message is displayed, and the text conversion process is aborted. While the file is loading, a requester is displayed that gives status information and offers the opportunity to cancel the text conversion.

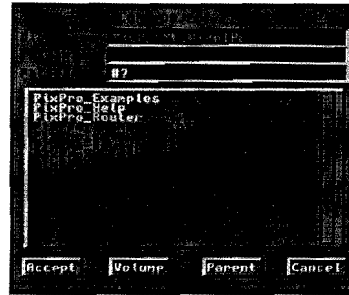
The text will be automatically converted into a flat 3D object (unless bitmap load config settings have been activated). The resulting object's size will be determined by the size of the original font. For example, if the font



was 100 points tall, the object will be 100 units tall. Different 3D programs use different measuring units; for example, one PixPro unit equals one meter in LightWave3D.

## 4.2 Loading a Bitmap Image (Keyboard Equivalent: I)

Click on the **Load** button (the same button used for loading objects) to load a bitmap or superbitmap. Any IFF bitmap image, using 32 colors or less will be recognized. While the file is loading, a requester is displayed that gives status information and offers the opportunity to cancel the operation.



The bitmap image will automatically be converted to a flat 3D object (unless bitmap load config settings have been activated). The resulting object's size will be determined by the width and height in pixels of the original bitmap. For example, if a 200-pixel wide by 100-pixel tall bitmap image is converted, the resulting object would measure 200 x 100 units in PixPro.

### \*\*\* NOTE \*\*\*

PixPro uses color zero (usually black) as the background color. Therefore, a logo that is black on a white background will present a problem because PixPro will want to make the white background the actual object rather than the logo you intend. The resulting object will resemble a white sheet with the logo cut out of it. Using an image conversion program such as ADPro or ImageFX, you can reverse the colors in the bitmap so this situation does not occur.

## 5.0 Undo (Keyboard Equivalent: u)

Clicking the **Undo** button will undo the last operation. This is useful if you have made an error and wish to restore your object. For **Undo** to function, you must have enough allocated memory for PixPro's Undo buffer. Without enough memory allocated, **Undo** is incapable of restoring your object to the previous state.



## 6.0 The View Window

The view window is the largest area on the PixPro screen. There are many ways of controlling the view.

### 6.1 View Mode Buttons (Keyboard Equivalents: F1 - F4)



Clicking on the **F** (Front) button (keyboard equivalent - F1) moves the View Window into **Edit Mode** and displays the current object from the front.

Clicking on the **T** (Top) button (keyboard equivalent - F2) moves the View Window into **Edit Mode** and displays the current object from the top.

Clicking on the **S** (Side) button (keyboard equivalent - F3) moves the View Window into **Edit Mode** and displays the current object from the side.

Clicking on the **Persp** (Perspective) button (keyboard equivalent - F4) moves the View Window into Perspective Mode where the object in memory can be viewed interactively with the mouse (see next section) from different angles and distances. The current object can be drawn in a variety of different combinations depending on the status of the Display buttons covered in section 6.5.

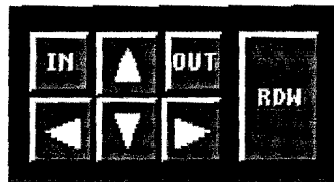
**Note:** While in the **Front**, **Top** or **Side** view, pressing the **a** key will auto scale the view window to fit the object. Also,

clicking the mouse anywhere in the **Front**, **Top** or **Side** view window and then selecting the **g** key will center that area in the view window. The **comma (,)** key will zoom the view in and the **period (.)** key will zoom the view out.

## 6.2 Mouse Controls

You can rotate the perspective view around the object by pressing and dragging the left mouse button from left to right and up and down. You can rotate around the Z axis by pressing and dragging the right mouse button. You can zoom in and out by pressing and dragging both mouse buttons up and down. While moving the mouse, the object will be replaced by a bounding box representing the dimensions of the object. The triangle at the bottom of the bounding box denotes the bottom of the object. When you let up on the mouse button, the object will reappear. Depending upon the complexity of the object, this may take a few moments. Clicking in the view window with either mouse button will break the redraw process then start again.

## 6.3 Control Buttons



Located at the middle bottom of the screen is a control button panel which influences the view of the object in the main view window.

If you are in the **Front**, **Top**, or **Side** views, clicking on the **right**, **left**, **up** or **down** arrow buttons will move the view

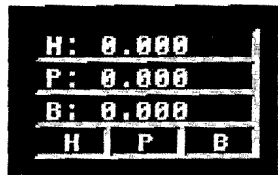
in the direction of the arrow. If you are in the **Perspective** view, clicking on the **right**, **left**, **up** or **down** arrow buttons will rotate the view around the center of the object. The keyboard equivalents for clicking on these arrow buttons are the arrow Keys below the **Delete** and **Help** Keys.

While in a view other than **Perspective**, each press of an arrow button or keyboard arrow will move the view one grid line. Hitting the **a** (the letter "a") key on the keyboard while in the **Front**, **Top** or **Side** views will automatically fit the object into the view window.

Clicking on the **In** and **Out** buttons will zoom the view toward and away from the object while in any view mode. The **Comma** (,) Key is the keyboard equivalent for zooming out from the object while the **Period** (.) Key will zoom in toward the object.

The large button marked **RDW** (Redraw) will redraw the screen after using the arrow buttons or the **In/Out** buttons for movement. The **Return** Key is the keyboard equivalent for the **RDW** button. While in **Perspective** view hitting the **RDW** button with the right mouse button will automatically reset any rotation values and fit the object into the view window.

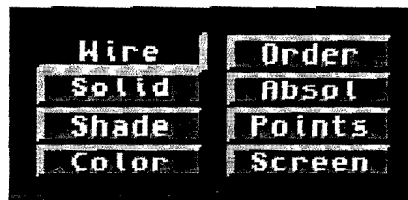
## 6.4 Coordinates



While in the **Front**, **Top** or **Side** views, the coordinates panel will show you the X, Y and Z position of the view. While in the **Perspective** view, the coordinates panel will show you the H (Heading), P (Pitch) and B (Bank)

coordinates of the view. Heading is rotation around the Y axis; Pitch is rotation around the X axis, and Bank is rotation about the Z axis. Pressing the 7, 8, or 9 Keys respectively on the numeric keypad will toggle each axis (X, Y, Z) on and off. Pressing the H, P and B buttons respectively will toggle each rotation axis (heading pitch and bank) on and off. If the axis button is turned off, movement or rotation will be constrained to the axes which are still turned on. Clicking the Left Mouse in any of the coordinate windows allows you to type in numeric values.

## 6.5 Display Mode Buttons



Near the bottom left of the PixPro screen are eight display mode buttons. These buttons control the current viewing mode of the view window. Certain buttons can only be selected while in certain view modes. Some buttons can be mixed to produce different types of viewing modes.

The **Wire** button will create a wireframe drawing of the object.

The **Solid** button displays a solid, hidden line removal, drawing of the object.

**Shade** creates filled, shaded polygons in grayscale.

Depressing the **Color** button adds the color of the polygons to any of the above modes.



The **Order** button shows only those polygons which are created (ordered) in a clockwise manner. This is extremely useful for LightWave, Videoscape and 3D Professional users.

The **Absolute** button displays the object with all its sides in their absolute or non-perspective size. You can still rotate the view around the object.

The **Points** button shows you the vertices or points of an object. You usually want this button selected when you are in **Edit** mode.

Pressing the **Screen** button causes all of the control panels to disappear and the view window to fill the whole screen. The **F10** and **ESC** keys are keyboard equivalents to toggle this function.

## 7.0 Object Editing Functions

One of the great features of PixPro is its ability to edit any object either created by it or loaded into it. The editing buttons are located in the bottom left of the screen.

Selecting either the **Top**, **Front** or **Side** view automatically puts you into edit mode. If you are in the **Perspective** mode, however, clicking on any editing button has no effect on the object.

### 7.1 Edit (Keyboard Equivalent: F6)

In this mode you can select and move points. To select a point, simply double click on it. Once it is selected, it becomes highlighted.

To move a single point, hold down the right mouse button ON TOP of it, then move the mouse. To move all selected points, hold down the right mouse button AWAY from all selected points and then move the mouse.



### 7.2 Pick (Keyboard Equivalent: F7)

**Pick** mode allows you to pick a group of points at one time. In **Pick** mode, you'll notice that your cursor becomes a little bounding box. Double clicking the left mouse button selects all points within the box (NOTE: You may need to position the box from two or three views). Holding down the right mouse button allows you to draw out a drag box, and all points that fall within the borders of the box become selected.

### 7.3 UnPick (Keyboard Equivalent: F8)

To deselect any points that are currently selected, click on the **UnPick** button and follow the same methods you used to pick them as described above.

### 7.4 UnAll (Keyboard Equivalent: F9)

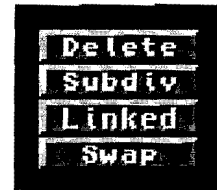
**UnAll** deselects ALL points that are currently selected. If no points are selected, **UnAll** will select all points.

### 7.5 Delete (Keyboard Equivalent: x)

The **Delete** button causes all the currently selected points to be deleted.

### 7.6 Subdivide (Keyboard Equivalent: v)

The **Subdivide** button adds a point to the edge of a polygon between two selected points which share a polygon edge. To add a point, select the two points on the targeted edge and simply click on the **Subdiv** button. This works for more than one segment selected. This function does not divide the polygon but rather the sides that make it up.



### 7.7 Linked (Keyboard Equivalent: c)

This function selects all the points that are connected via polygon edges to the currently selected points.

### 7.8 Swap (Keyboard Equivalent: b)

Click on the **Swap** button to swap selected with unselected points and vice versa.

—

—

—

## 8.0 Object Manipulation Functions

These functions allow you to manipulate your flat objects in a variety of ways. **Smoothing** can be applied to an object to help get rid of 'jaggies' or stair steps in an object. Objects can be **extruded** to give them depth. **Beveling** will create beveled edges around an object and objects can be **spun** around a center to create new objects.



### 8.1 Smooth (Keyboard Equivalent: 1)

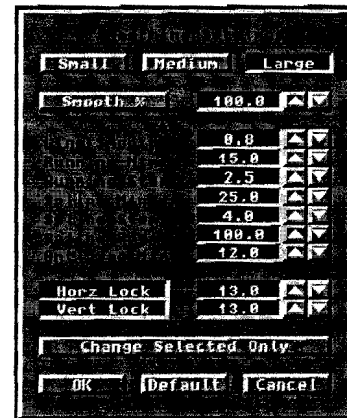
Clicking the **Smooth** button brings up the **Smoothing Settings** requester, where you can set values to eliminate extraneous points, resulting in smoother, cleaner objects.

The three buttons at the top of the requester set the various **Smoothing Settings** to useful defaults:

Clicking on **Small** sets all values to defaults appropriate for a small change in smoothing. This is good starting point for imported text 40 points or less in height.

Clicking on **Medium** sets all values to defaults appropriate for a medium change in smoothing. (40-80 points high)

The **Large** button sets all values to defaults appropriate for a large change in smoothing. (80-120 points high)



(Note: all units are measured in pixels)

The **Smooth Percent** field allows you to set a variable percentage of smoothing. You can click in the field to erase the current value and then enter a new value, or you can click on the up and down arrow buttons to increase or decrease the value. Once this value is changed, the **Small**, **Medium** and **Large** buttons have no effect on the object.

**Line Smoothing** will attempt to eliminate extraneous vertices (points) leaving a smoother, cleaner set of fewer vertices. Using the **Line Smooth** field value as the greatest amount of deviation, this function throws away vertices that do not deviate more than this value from segment to segment. You can click in the field to erase the current value and then enter a new value, or you can click on the up and down arrow buttons to increase or decrease the value.

Imagine lining a ruler up against a stair-stepped line in a bitmap. PixPro measures the points on each side of the ruler, and if they are closer than the **Line Smooth** value away from the ruler, they will be eliminated when line smoothing is applied. (See pic 1 on page 59)

**Line Running** attempts to look at segments (a segment goes from point to point) of the polygons and attempt to plot a smooth course through them. The **Running Min** field determines which segments will be included in the line running algorithm. Any segments that are greater than the **Running Min** value will NOT be included in the line running procedure. Line Running starts at the first point in the polygon and measures the distance to the next point. If that point is less than the **Running Step** value away from the first point, line running keeps measuring to the next point. As soon as a point falls outside of the **Running Step** value, a line is drawn along the average path of all the points that are within this measurement and the points are



deleted. The length of the new segment is equal to the **Running Step** value. You can click in the field to erase the current value and then enter a new value, or you can click on the up and down arrow buttons to increase or decrease the values. (See pic 2 on page 59)

**Spline Fitting** allows splines (curved lines) to be calculated between joined segments. For example, this function is very useful in fitting a spline to certain portions of a logo. The process will not affect any segments that are larger or equal to the **Spline Min** field value. **Spline Fitting** then goes through the remaining segments, measuring the angle between two segments that are joined (both segments must be smaller than the **Spline Minimum** value). If this angle is greater than or equal to the **Spline Angle** field value (see below), a spline is calculated to pass through the segments forming the angle. The number of resulting vertices is determined by the **Spline Step** field value (see below).

The **Spline Step** field value determines the number of vertices created by spline fitting. The spline formed has a vertex every **Spline Step** number of units. The smaller the Spline Step value, the more points on the spline.

The **Spline Angle** field determines the angle between two segments at which a spline is calculated in spline fitting. If the angle measured is less than the **Spline Angle**, no spline will be plotted through the segments (see **Spline Min** above for more details). (See pic 3 on page 59)

The **Anti Jaggie** function is useful in eliminating "jaggies" from your bitmap. The **Anti Jag Minimum** field value determines the length of horizontal or vertical segments that will be preserved while applying the anti-stair stepping algorithm. If you have four vertices that stair-step down and to the right, this function plots a new, 5th point exactly in the middle of points 2 and 3. Points 2 and 3 are deleted

resulting in a smoothed line with points 1, 5, and 4.

When toggled on, the **Horz Lock** button causes all horizontal segments that are greater than or equal to the **Horz Lock value** to be preserved and NOT incorporated into any of the above algorithms.

The **Vert Lock** button and value field preserves all vertical segments in the same manner as the **Horz Lock** option. You can click in the field to erase the current value and then enter a new one, or you can click on the up and down arrow buttons to increase or decrease the value.

In PixPro, a polygon can be selected or not selected. A polygon or portions of a polygon can be considered selected when some or all the vertices that make up the polygon are selected. In the **Smoothing Settings** requester, the **Change Selected Only** button allows changes to be applied ONLY to the selected portions of the polygon or polygons. When the **Change Selected Only** button is toggled off, changes are applied to the entire object in memory.

It is important to note that whenever PixPro executes a function with **Change Selected Only** on, all new vertices created appear selected. When **Change Selected Only** is off, new vertices are created deselected.

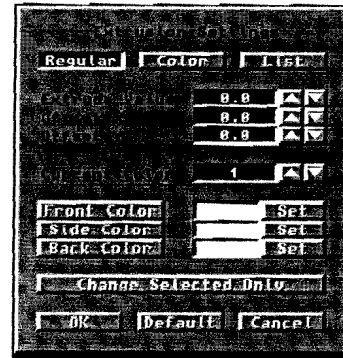
The **OK** button performs the selected **Smoothing** function on the data in memory and returns the user to the PixPro main screen. The **Default** button restores the settings in the **Smoothing** requester to their original default conditions. The **Cancel** button returns the user to the PixPro main screen without executing any smoothing functions.

## 8.2 Extrude (Keyboard Equivalent: 2)

Clicking the **Extrude** button brings up the **Extrusion**

**Settings** requester where you can set values to extrude objects. The three buttons at the top of the requester set alternatives for controlling extrusions:

The first time you enter the **Extrusion** requester in a session, the default mode is **Regular**. In this mode, the object is extruded according to the **Extrude Value** (below), measured in pixels. For example, an object 100 pixels wide, with an Extrude Value of 100, gives you a perfect square when viewed from the top in **Edit** mode.



The **Color** mode extrudes the current object based on the intensity of the colors of the object's polygons. If you enter an **Extrude Value** of 100, a pure white polygon would be extruded the full 100 pixels; a medium gray polygon (50% white) would be extruded half of the **Extrude Value** while a dark colored polygon is the least extruded.

**List** mode lets you go through all colors and assign each of them a specific value for the extrusion. You use the **Current Color** gadget (below) to cycle through Colors 1 to 31. Each color can be assigned its own **Extrude Value**, **Offset X**, **Offset Y**, **Front**, **Side**, and **Back** Colors.

**Note:** When performing a **List** extrusion, you should not alter the colors of the object's polygons before performing the extrusion. PixPro uses the original color of the bitmap to base its **List** extrusion values on.

The **Extrude Value** determines the amount in pixels that the object is extruded along the Z axis. You can click in

the field to erase the current value and then enter a new one, or you can click on the up and down arrow buttons to increase or decrease the value.

The **Offset X** value controls the amount of horizontal skew you give to the back polygons of the object in extrusion.

The **Offset Y** value controls the amount of vertical skew you give to the back polygons of the object in extrusion.

The **Front Color** button, if selected, indicates that the color shown in the box next to it will be used in the front polygons of the extrusion. To change the color, click on the **Set** button to bring up the **Change Color** requester where you can adjust the R, G and B sliders to set a new color which is reflected in the color box. Click on **OK** to return to the **Extrude** requester with the **Front Color** set. Clicking **Restore** restores the settings in the **Front Color** requester to their last used conditions and clicking the **Cancel** button returns the user to the **Extrude** requester without changing the color.

In **Regular** mode, changing the **Front Color** changes all front polygons to that color, even if they differ in color.

In **Color** mode, the extrusion takes place based on the original colors, but the extruded object will have all front polygons set to the defined **Front Color**.

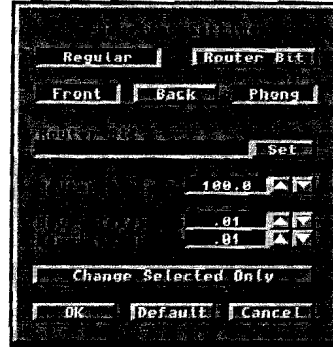
The **Side Color** button, if selected, indicates that the color shown in the box next to it will be used in the side polygons of the extrusion. To change the color, use the same methods described for **Front Color** (above).

The **Back Color** button follows the same principles as both **Front** and **Side Colors** (above).

The **Change Selected Only** option in the **Extrude Settings** requester will extrude only the selected parts of the object.

### 8.3 Bevel (Keyboard Equivalent: 3)

Clicking the **Bevel** button brings up the **Bevel Settings** requester where you can set values to add a controllable bevel to extruded bitmaps and objects. The beveled look is ideal for adding a professional gloss to logos, titles and animations.



The **Bevel** requester defaults to **Regular** mode. In this mode, you select any combination of **Front**, **Back**, and **Phong** buttons, set the **Bevel Inset** and **Height** values and execute the function on the selected polygons or object.

**Router Bit** mode allows you to utilize a user-defined polygon to shape your bevel. Selecting the **Router Bit** button allows you to assign a router bit to control your **Bevel** function. In this mode, you select any combination of **Front**, **Back**, and **Phong** buttons, choose a **Router Bit** File and set a **Router Scale** value, then execute the function on the selected polygons or object. The **Bevel Inset** and **Height** buttons are ignored when using router bits (see the Router Bit field and Router Scale discussions below for more information).

With the **Front** button selected, the front of the current object in memory will be beveled. With the **Back** button selected, the back of the current object in memory will be beveled. When the **Phong** button is depressed, the polygons that make up the bevel will have their own set of

vertices. This allows you to phong shade your object in a renderer, keeping the edge between the front face and the bevel sharp, while smoothing the bevel and the rest of the object. With **Phong** not selected, the bevel polygons are integrated with the front polygon or polygons.

Click on the **Set** button to open the file requester to choose your **Router Bit** File. All router bits must be one polygon only and must be saved in **Router Bit** format. The actual bevel portion will follow the outline of the right edge of the bit's polygon (when seen from the front view). PixPro finds the upper-right and lower-right most points and follows the path between those points as the router bit. Note that the router bit polygon must be in clockwise order. You can confirm this by viewing the bit in Perspective mode from the front with **Shade** and **Order** turned on. (See above for Shade and Order information).

The **Router Scale** field value can be adjusted to alter the size of the router bit for use in the Bevel function. You can click in the field to erase the current value and then enter a new one, or you can click on the up arrow to increase the value or on the down arrow to reduce it.

The **Bevel Inset** value, measured in units from the edge of the front polygon to the edge of the rest of the object, controls how far in a bevel goes on the current object. Once again, you can click in the field or use the arrow buttons to adjust the value.

The **Bevel Height** field controls how far out the bevel extends from the current object. The bevel is actually added to the selected polygon or object. For example, if you have a logo that is 50 units deep, and you set a **Bevel Inset** of 5 and a **Bevel Height** of 10, the resulting object will be 60 units deep.

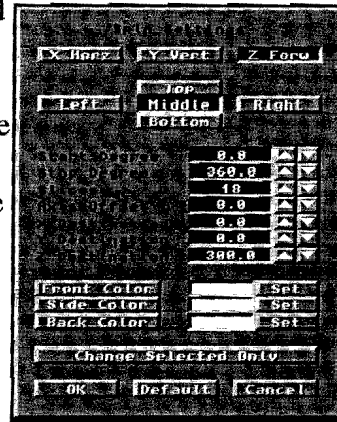


The **Change Selected Only** option in the Bevel Settings requester will cause the bevel to be applied only to selected polygons.

#### 8.4 Spin (Keyboard Equivalent: 4)

Clicking the **Spin** button brings up the **Spin Settings** requester where you can take a converted bitmap and spin it on a defined axis for a defined number of degrees.

The **X Horz** button defines the X or Horizontal axis as the **Spin** axis and makes available the **Top**, **Middle** and **Bottom** Justification buttons beneath it. The default is the **Middle** button.



Clicking on the **Top** button will place the spin point at the top of the object (the highest point coordinate). Clicking on **Middle** will place it in the middle (the center point coordinate), and clicking on **Bottom** will place it at the bottom of the object (the lowest point coordinate).

The **Y Vert** button defines the Y or Vertical axis as the **Spin** axis and makes available the **Left**, **Middle** and **Right** Justification buttons beneath it. The default is again the **Middle** button.

Clicking on **Left** will place the spin point at the left edge of the object. Clicking on **Middle** will place it in the middle and clicking on **Right** will place it on the right edge. The **Z Forw** button defines the Z or Forward axis as the **Spin** axis and makes available the **Top**, **Left**, **Middle**,

**Right** and **Bottom** Justification buttons beneath it. **Middle** is the default.

As in previous examples, clicking on **Top**, **Left**, **Middle**, **Right** and **Bottom** will place the spin point at the top, left, middle, right and bottom edge of the object respectively.

**Start Degree** is the angle at which the spin will start; **Stop Degree** is the angle at which the spin will stop, and **Slices** control the number of segments that make up the final spun object.

The **Axis Offset** field will offset the spin point the number of pixels entered as its value, in the direction of the axis selected. For example, if **Y Vert** is selected, the offset will be on the X axis. A negative value would move the spin point of the object to the left, a positive value would move it to the right. If **Middle** is selected for the spin point, this value will have no effect.

**X Destination** is where the end of the spin will stop in the x plane. **Y Destination** is where the end of the spin will stop in the y plane, and **Z Destination** is where the end will stop in the z plane. Think of **Z Destination** as how "deep" the spin goes.

The **Front**, **Back** and **Side Color** buttons in this requester operate the same way as they do in the Extrude requester only their results are applied to the Spin functions you are working with.

The **Change Selected Only** option in the **Spin Settings** requester will only spin the selected portions of the object.

## 9.0 Data Settings

Clicking on the **Data** button will open the **Data Settings** requester, this requester lets you view information about the current object in memory and allows interactive modification of the data that makes up the object. A set of powerful object optimization algorithms includes functions that reduce redundant data, increasing rendering speed and assisting in converting objects from one file format to another. Multiple values can be set, while changes must be applied one at a time.



LightWave 3D works best with many-sided polygons, but will accept three-sided polygons like those created in Imagine and Sculpt. These triangle-based objects do not order their vertices and contain no direction information. In order to render these objects in LightWave, without conversion, you need to click on LightWave's **Double-Sided** button, which increases rendering time. The following functions help to optimize your objects before you save them into their new formats:

### 9.1 Polygon Flipping

The **Flip** button of the **Data Settings** requester will intelligently attempt to flip or re-rotate polygons into clockwise order, so they will display correctly when moved from triangle-based Imagine, Turbo Silver and Sculpt formats to LightWave, VideoScape and 3D Pro formats.

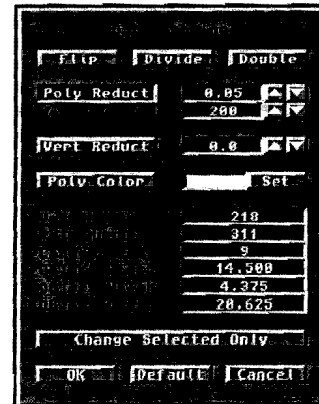
While Flipping is being performed, a status requester displays values indicating the number of linked or separate pieces that have been found, also, how far it has progressed

on that piece.

Polygon Flipping is intelligent, but it is not fool-proof. It is algorithmically impossible for random polygons to be put into the proper order of rotation consistently. This is true because not all polygons form legal solids. If just one polygon makes up the object, the direction of that polygon depends on the point of view. Polygon Flipping will attempt to look at a polygon first from the front, then the top, then the side and will put the polygons in order from those points of view.

## 9.2 Polygon Division

The **Divide** button takes polygons of more than three sides and splits them up into triangles. This is useful for breaking up polygons with too many sides which aids in solving quirky rendering problems. **Polygon Division** is automatically performed on multi-sided polygons when objects are saved in Imagine, Turbo Silver and Sculpt formats.



**Polygon Division** does not work very well, if at all, on polygons that have no width at any point. The same is true for crossed polygons, like a twisted square. PixPro's **Polygon Division** loves to work on normal, healthy polygons, of any size.

## 9.3 Polygon Doubling

Clicking on the **Double** button creates a duplicate of every polygon in the current object, building these new polygons in the opposite direction from the originals. If **Polygon Flipping** has not completely solved display problems, try **Doubling** before saving to VideoScape or 3D Pro formats. With Lightwave 2.0 or greater, it is more efficient to use Lightwave's Double-Sided button, then use **Polygon Doubling**, but still try **Polygon Flipping** first.

#### 9.4 Polygon Reduction

The **Poly Reduct** function reduces and rebuilds polygons. The algorithm searches for polygons that are coplanar (on the same plane), deletes them and rebuilds them as one larger polygon. For example, a cube built of triangles in Imagine consists of 12 polygons, pairs of which are coplanar on each face of the cube. **Polygon Reduction** rebuilds the cube with 6 square polygons. **Polygon Reduction** also searches for vertices that do not add to the definition of a polygon; a vertex exactly in the middle of one side of a square is considered redundant and is deleted.

The value in the **Polygon Reduction** field represents the threshold below which two non-coplanar polygons will be considered as coplanar and therefore Reduced. In other words, if the angle of difference in degrees is less than the default .05, then the two polygons are remade as one. Most values are less than 1.0 degree, although hand made objects may have few coplanar polygons and may require higher values. You can click in the field to change or enter a new value or you can use the arrow keys to increase or reduce it.

The **Max Poly Sides** field tries to create the most efficient polygon possible dependent on the Poly Reduct value. **Max Poly Sides** sets the maximum number of sides for polygons

after reduction. This is useful for specialized file format conversions where a specific number of sides is required.

## 9.5 Vertex (Point) Reduction

The **Vert Reduct** field eliminates redundant or nearly redundant points or vertices. If two vertices are located at the same spot only one is necessary to preserve the object's shape. **Vertex Reduction** senses this and deletes the extra point.

The value in the **Vertex Reduction** field represents the threshold below which two closely-located vertices will be considered redundant and therefore Reduced. Any distance equal to or below the value will cause the vertex to be deleted; any vertices further apart than the value will be left alone.

The size of the object determines the optimal value for **Vertex Reduction**. Look at the object **Width**, **Height** and **Depth** values in the Data Settings requester to get a feel for what value should be used. It is best to start with a small value and increase it as needed. If the dimensions of an object were all 1, a good value might be .0001.

## 9.6 Polygon Color

You can use **Poly Color** to change the color of either selected polygons or the entire object in memory. To change the color of a polygon, select all the vertices that make up the polygon, then enter the **Data** requester and depress the **Change Selected Only** button. Notice that the current color of the selected polygon is loaded into the color box to the right of the **Poly Color** button. Also the object width, height, depth, etc. fields now reflect the

selected polygon(s). Click on the **Set** button to bring up the **Change Color** requester. You can adjust the R, G and B sliders to set a new color, which is reflected in the color box. Click on **OK** to return to the **Data** requester with **Poly Color** set. Clicking **Restore** restores the values in the color requester to their last saved conditions, and clicking the **Cancel** button returns the user to the **Data** requester without changing the color.

If the **Change Selected Only** button is deselected, all polygons of the current object will take on the new color.

## 9.7 Selected Data

The following windows display information about the selected polygon(s):

The **Vert Number** value represents the number of current vertices.

The **Poly Number** value represents the number of current polygons.

The **Largest Poly** field represents the number of sides on the largest polygon.

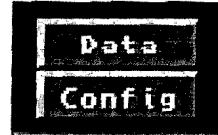
The **Object Width** is the actual width of the current object.

The **Object Height** is the actual height of the current object.

The **Object Depth** represents the actual depth of the current object.

## 10.0 Configuring Bitmap Load Settings

Clicking the **Config** button brings up the **Bitmap Load Settings** requester where you can configure PixPro to apply various functions to a bitmap (which includes text) whenever a bitmap is loaded.



### 10.1 Color

Selecting the **Color** button tells PixPro to use the bitmap's colors for extrusion purposes when loading a bitmap. In the case of text, this defaults to white.

### 10.2 Monochrome

Selecting the **Monochrome** button tells PixPro to ignore the bitmap's colors when loading. PixPro will treat the bitmap as if it was a two color image. Color 0 is the background color, and any other colors are treated as if they were all color 1.

### 10.3 Grid

Selecting the **Grid** button tells PixPro to use the bitmaps' color intensity to create a grid map of the bitmap. This map of square polygons, will be as wide and as tall as the screen or brush that the bitmap was saved as. The high and low points of the grid map are determined by how bright the colors of your bitmap are. PixPro looks at the bitmap at



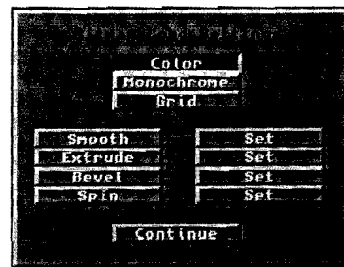
every intersection of the grid which is laid on top of the bitmap. The average color intensity at that intersection on the grid determines how high that point of the grid will be. The highest and lowest points of the grid will fall between 0 and the value specified in the **Config Extrude Value** requester, which is found via the **Config** requester. The width and height of each square that make up the grid are defined using the **X Offset** and **Y Offset** values also found on the **Config Extrude** requester.

Note that you can use the spray paint tool in Deluxe Paint to produce interesting bitmaps that you can then convert using the **Config Grid** setting. It is also best to import brushes, rather than full screens, since the size of the resultant grid object is controlled by how large the bitmap is.

#### 10.4 Smooth/Extrude/Bevel/Spin Settings

These four buttons allow you to set up default **Smooth**, **Extrude**, **Bevel** and **Spin** settings that will be used on any image or text that is loaded into PixPro. Each button brings up a corresponding

requester screen. The values you set are preserved separately from the ones in the main screen. For instance, click on the **Set** button to the right of the **Smooth** button to open the **Config**

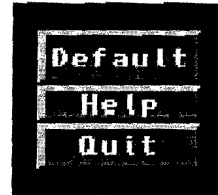


**Smooth Settings** requester. When loading a bitmap, the values you set will be applied and the data will be preserved under this requester. Follow the same procedure for each of the other three options. Config settings will be applied to the bitmap upon loading if any or all of these four buttons are selected.

Click on the **Continue** button to accept the settings and exit the **Bitmap Load Settings** requester.

## 11.0 Setting Defaults

Clicking the **Default** button found in the lower right hand of the screen brings up the **Default Settings** requester where you can choose from the following options:



### 11.1 System Memory

Clicking the **Memory** button brings up the **Memory Allocation** requester where you can control the amount of memory PixPro will allocate. This type of allocation allows you to obtain the fastest rendering and bitmap creation possible. PixPro requires that memory be allocated before any type of work begins.

At startup, Pixpro will attempt to allocate memory for 22,500 vertices and polygons. This default can be changed by clicking on the up and down arrows in the requester. The following values are displayed: **Maximum Vertices**, **Maximum Polygons**, **Bytes Required**, and **Bytes Remaining**. Clicking on **OK** accepts the new values, while clicking on **Cancel** returns you to the **Default** requester without any changes.

### 11.2 Text Color

Clicking the **Text Color** button brings up the **Change Color** requester where you can adjust the R, G and B sliders to set a new color, which is reflected in the color box. The **Text Color** is used throughout PixPro as the color for button text, cross-hairs and selected vertices in

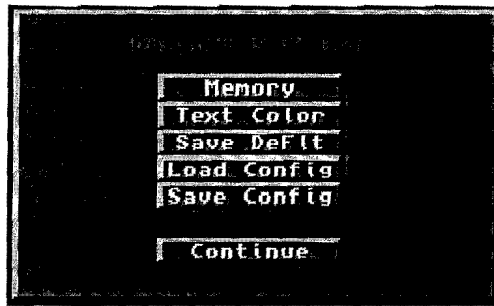
**Edit mode.**

Click on **OK** to return to the **Default Settings** requester with the **Text Color** set. Clicking **Restore** returns the settings in the **Text Color Settings** requester to their last saved conditions, and clicking the **Cancel** button returns the user to the **Default Settings** requester without changing the color.

### 11.3 Save DeFault

Clicking on the **Save DeFlt** button takes all the current settings in ALL requesters and saves them in the S directory of your System as

the **PixPro\_Defaults** file. When you start PixPro the program looks in the S directory for **PixPro\_Defaults** and loads the program with those settings.



**Note:** This is a convenient way to set up default paths for loading and saving objects. Simply select the paths in the Load/Save requesters **WITHOUT** entering a filename in the filename string gadget (make sure the last directory has a "/" at the end of it).

### 11.4 Load and Save Configurations

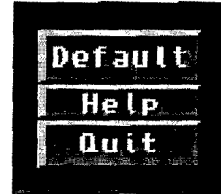
Clicking the **Load Config** button brings up the **Load Configuration File** requester where you can load custom configuration files you have previously saved with **Save Config**. Keep in mind, however, that **Load Config**

changes ALL the values in ALL requesters, so you may change the values in one requester when you load in values for another function.

## 12.0 Help (Keyboard Equivalent: Help Key)

Clicking on the **Help** button in the lower right of the screen will perform two functions:

1. If you click on it once and then click on the button you want help with, a help requester will pop up and give you information about the selected item.



2. If you click once on the help button and then click on it a second time, an **About** screen will appear and give you information about the program.

## 13.0 Quit (Keyboard equivalent: q)

Clicking the **Quit** button in the lower right hand corner of the screen will exit out of PixPro, first asking to make sure you really want to quit.

## 14.0 Appendices

### 14.1 Keyboard Shortcuts

**F1** Front View Mode  
**F2** Top View Mode  
**F3** Side View Mode  
**F4** Perspective View Mode

**F6** Edit  
**F7** Pick  
**F8** UnPick  
**F9** UnAll  
**F10** Toggle Full Screen View on-off

**1** Smooth  
**2** Extrude  
**3** Bevel  
**4** Spin

**7** X Button Toggle on-off (numeric keypad)  
**8** Y Button Toggle on-off (numeric keypad)  
**9** Z Button Toggle on-off (numeric keypad)

**h** H Button Toggle on-off (**Persp** view only)  
**p** P Button Toggle on-off (**Persp** view only)  
**b** B Button Toggle on-off (**Persp** view only)

**ESC Key** Toggle Full Screen View on-off  
Cancel from any requester

**SpaceBar** Toggle Through Default Display Modes



- a** Auto Fit Object into Window (F, T or S view only)
- b** Swap
- c** Linked
- g** Go to center
- v** Subdivide
- x** Delete
- ,** Zoom Out
- .** Zoom In

**Arrows** Scroll in Direction of Arrow One Grid

Both Mouse buttons held down will zoom in and out while in Perspective Mode.

## 14.2 Tips and Tricks

### Phong Extrusion

When you extrude, PixPro is automatically set up for a "phong" type of extrusion. This means a separate set of points are created for the front and for the sides. If this is not the way you would like PixPro to work, you can go into **Data Settings** and use the point reduction feature.

### Lone Points

By design, PixPro does not like lone vertices. If a vertex is not on a polygon (referred to as "being used"), it will be loaded and displayed, however, for certain applications such as extruding or smoothing, those lone vertices will be eliminated.

### Beveling

Beveling usually finds the front faces and bevels them because these faces have already been extruded. In that sense, there is no need to preserve the old front face since you are building a new one. The original face that the bevel is based on is deleted. If you have not extruded your logo, the Beveling will work, but you need to be aware that the resulting object will not have a back face.

When importing objects from other modelers, it is important to know how PixPro looks at polygons. If you have ever noticed how beveling works on a polygon seam (where a polygon joins onto itself, like a B or a P built out of one polygon), the bevel does not intrude onto the seam.

So, you can have 1 polygon representing the letter O, but there will be a seam. When you bevel such a polygon, you want the bevel on the outsides and insides but not on the seam. PixPro realizes that it is a seam because the edges that represent the seam share the same vertices. If the edges of the polygon at the seam each had their own vertices, PixPro would not recognize that as a seam (just because the polygon jumps back on itself forming a seam does not mean that it is a seam... points must be shared). For example, a good way to determine if there is a valid seam is to move one of the vertices on the seam. If the whole seam moves, everything's great. If just one side of the polygon moves, there were obviously two points in the same place, and that does not constitute a seam. It is important to know what is a seam and what isn't when building your own objects.

Beveling will only work on polygons that are perfectly 2D; in other words, the polygons must be in the X,Y plane only. When you look at these polygons from the side or top, there is no depth; you see just a vertical or horizontal line. If you try to bevel a slew of angled polygons, it just won't work!

### **Router Bits**

It's easy to create properly sized router bits by creating a 2x2 pixel brush and then converting that in PixPro. If you go into **Edit** mode, and then you subdivide the right side to add detail, you know that your router bit is 2 pixels high by 2 pixel wide compared to your converted logos.

## Selected Polygons

In PixPro, a polygon can be selected or not selected. If ALL the vertices making up the polygon to be selected are selected, then the polygon is selected. The following functions use this terminology for determining which polygons to effect when **Change Selected Only** is depressed: **Data Requester**, **Bevel Requester**. The other three, **Spin**, **Smooth**, and **Extrude** will affect just portions of a polygon, so these functions don't work on the idea of having a whole polygon selected. For example, you can extrude just half of a polygon, but you cannot bevel just half of a polygon, nor can you divide using the **Data** requester just half of a polygon. If half of a polygon is selected and you try to bevel or use the **Data** requester, nothing will happen to that polygon.

PixPro now has no maximum polygon size. When it saves out to VideoScape or LightWave or any other multi-sided polygon format, it will break any polygons that are larger than 200 sides up into 200 sided polygons.

Hint: If you have a large polygon and want to break it up into slightly smaller polygons, not triangles, use Divide in the data requester and then use Poly Reduction with a **Max Poly Side** value of what ever you want.

## Using PixelPro with the Video Toaster

PixPro is a great Toaster tool. Capturing a client's logo with the Toaster and turning it into a 3D object so you can use it in LightWave is very easy. Simply capture the logo with a camera or better yet, a scanner. and save it in ToasterPaint as an RGB (24 bit IFF).

Next, load the RGB image into an image conversion

program such as ADPro or ImageFX and convert it down to a resolution that PixelPro can understand (32 colors or less). Generally, if you just need the shape of the logo for your work, convert the image down to 2 colors.

Once the image is converted, you can easily load it into PixPro to manipulate.

Note: Most logos scanned into the Toaster are going to end up as black text on a white background (like a business card). Remember that this will not convert the way you think it will so the image needs to be reversed so it is white on black. Use a paint program or reverse the image in your image conversion program.

## 14.3 Notes about Object Formats

### 14.3.0 LightWave 3D

PixPro will both load and save LightWave 3D V1.0 & V2.0 object files. PixPro will load and save vertices, polygons and colors. Each color saved is given a name and can be seen in LightWave 3D's surfaces requester. When saving an object converted from text, you will be prompted as to whether you wish to scale the height of the object to 1 meter. This is useful for standardizing your logos sizes in LightWave

### 14.3.1 3D Professional

PixPro will both load and save 3D Professional V1.0 object files. PixPro will load and save vertices, polygons and colors.

### 14.3.2 Imagine/Turbo Silver

PixPro will both Load and Save Imagine V1.0 & V2.0 and Turbo Silver V3.0 and greater object files. It will load vertices, polygons, colors and recognize Imagine grouped objects. PixPro will save vertices, polygons, colors and will name each separate piece of the object. These names will be visible in the Choose By requester portion of Imagine's Detail editor. These pieces are not determined by the original grouped structure but are determined by finding individual pieces of an object. **Polygon Division** is performed automatically before saving to these formats.

### **14.3.3 AutoCAD DXF**

PixPro will load and save AutoCAD DXF object files, a subset of entities and save 3D Faces with no colors.

### **14.3.4 WaveFront**

PixPro will load and save Wavefront object files, vertices and polygons, but no colors.

### **14.3.5 Digital Arts**

PixPro will both load and save Digital Arts' object files. PixPro will load and save vertices and polygons, but not colors.

### **14.3.6 Videoscape BINary/Videoscape ASCII**

PixPro will both load and save VideoScape V1.0 & V2.0 ASCII and Binary object files. It will load vertices, polygons and colors, save vertices, polygons and the proper color codes that match the current object in memory colors. PixPro saves to the VideoScape file format using colors from the bitmap or 3D object that most closely match those found in VideoScape's fixed palette.

### **14.3.7 Sculpt 3D**

PixPro will both load and save Sculpt 3D/4D object files, PixPro will load and save vertices, polygons and colors. **Polygon Division** is performed automatically before saving to this format.

#### **14.3.8 Draw 4D/Draw 4D Pro**

PixPro will both load and save Draw 4D, Draw 4D Pro object files. PixPro will load and save vertices, polygons and colors.

#### **14.3.9 Caligari**

PixPro will both load and save Caligari object files. PixPro will load vertices, polygons and colors, but will not load holes. PixPro will save vertices, polygons and colors.

#### **14.4.0 Router Bit**

PixPro will allow you to save out a 1 polygon object as a PixPro router bit for use with PixPro beveling. The format used to save the router bit is the VideoScape ASCII format.

#### **14.4.1 Vista Pro**

PixPro will allow you to load only a Vista Pro DEM file. A grid is constructed and represents the information contained within the Vista Pro DEM file.

#### **14.4.2 Scenery Animator**

PixPro will allow you to load only a Scenery Animator DEM file. A grid is constructed that represents the information contained within the Scenery Animator DEM file.



## 15.0 Tutorials

### 15.1 Converting text into an object

1. Insert the PixelPro disk into your floppy drive and start the PixelPro program.
2. Click on the **Text** button.
3. When The **Enter Text to Convert** requester appears, Type ABC (use capitals). Click on the **OK** button when you are done.
4. When the **Font** requester appears, it will default to your **Fonts:** directory. Change the Directory name so it reads **DF0:Fonts**. In this directory, you will find a Palacio drawer. Double click on the Palacio drawer and you will see the font sizes. 121 is the only font size in this directory. You can either click once on 121 and then click **Accept** or you can double click on 121.
5. The font will automatically be converted into a 3D object and will be sized to fit into the Perspective view.
6. Using the left mouse button, rotate the view around to see your object from different angles. Click on the **RDW** button with the right mouse button to reset the view so it is looking straight at the object.
7. Click the Spacebar to cycle through different display modes. Clicking on the **RDW** button will redraw the object with the new modes.
8. Select **Wire** mode and click the **Smooth** button to bring up the **Smoothing** requester.

9. Make sure the **Large** button is selected and leave all the values at their defaults. Click on the **OK** button to start the smoothing process.

10. Notice how much smoother the text appears! Let's go in and edit some points.

11. If you look at the left hand leg of the A, you will see that the bottom of the foot has lost some of its form. Click on the **F** button to go into Front view and **Edit** mode.

12. While in **Edit** mode, click the left mouse button over the left hand leg of the A. Now hit the **g** key (Go to center) to send that area to the center of the screen. Click on the **period** key a few times to zoom the screen in and then click on the **RDW** button.

13. Now, double click on top of the point that needs to be moved. The point will become selected. Holding down the right mouse button anywhere on the screen and moving it will move the point. To deselect the point, just double click on it again. Repeat for other points that should be moved. Remember, **Undo** will undo your last operation if you make a mistake.

14. To select multiple points at one time, click on the **Pick** button to enter **Pick** mode. If you Double click anywhere now, you will select all points that are within the select box. If you use the right mouse button, you can draw a bounding box area around any points you wish to select. To move these selected points, you must go back into **Edit** mode and move them with the right mouse button.

15. Once you have moved points around, click on **Unall** to deselect all points and then click the **A** key to resize the view to fit the object.

16. Next, let's extrude the current object. Click on the

**Extrude** button to open the **Extrusion** requester.

17. Click on the **Regular** button and set the **Extrusion Value** to 30. Next click on the **Front Color Set** button and set the front color to a color of your choice. Make sure the **Front Color** button is selected so this change will take place. Repeat this process for both the **Side** and **Back** colors.

18. Click on the **OK** button to accept these values and extrude the object.

19. Go into **Perspective** mode and view the object from a few different angles. Use the spacebar to cycle through a few different default view modes and click on the **RDW** button to see each one.

20. Finally, let's give a beveled edge to the front of the object. Click on the **Bevel** button to open the **Beveling Settings** requester.

21. Make sure that **Regular** and **Phong** are set, and set the **Bevel Inset** and **Bevel Height** to 1.5 each. Click on **OK** to accept these values.

22. Congratulations, you now have an extruded, beveled object that you can save out and take into a 3D program for rendering.

## 15.2 Converting a bitmap into an object

1. Insert the **PixelPro** disk into your floppy drive and start **PixelPro**.

2. Click on the **Load** button or press the **l** key to open the **Load** requester. Select the **FakeCo.bitmap** image located

in the **PixPro\_Examples** drawer of the PixPro disk. Once the image is selected, click on the **OK** button to automatically convert the image into an object. View the image in Perspective view using the mouse.

3. Once the image is converted, click on the **F** button to enter the **Front** view. Notice how jaggy the diagonals and curves of the logo are.

4. Next, click on the **Smooth** button to access the **Smoothing** requester. Make sure the **Large** button is selected and leave all of the other values at their defaults. Click on **OK**.

5. Notice how extraneous points have been deleted in the logo and the jaggies that were on the diagonals and curves are now gone. Click on the **Extrude** button to access the **Extrusion Settings** requester.

6. Click on **List** and deselect (if they are selected) the **Front**, **Side** and **Back Color** buttons.

7. We are going to extrude the triangles to different depths than the logo. In order to do this, you need to know which colors are used in the bitmap. The FakeCo text uses color no. 2 and the triangles use colors no. 4, 5, 6 and 7 respectively from left to right. (This can be seen if you load the image in Deluxe Paint and look at the color registers.

8. Make sure the **Current Color** is set to 2 ( the color register for the logo) and type in an **Extrude Value** of 50. Cycle through the **Current Color** settings and set colors 4, 5, 6 and 7 **Extrude Values** to to 50, 60, 70 and 80 respectively. Click on **OK** to accept your changes.

9. That's all there is to it! After the image has been

extruded, try looking at it from different views to see how the different parts of the bitmap image have been extruded to different depths.

10. Select Save and choose an output format if you wish to save this object out.

## 16.0 Technical Support and Q&A

Of course, if all else fails, there's always technical support. We do ask one thing before you dial our number, and that is that you quickly scan the below Q&A section to see if your problem is talked about, if not, (sigh) dial away.

**AXIOM TECHNICAL SUPPORT: 612-888-0352**  
Hours 9am-5pm Central

**Q:** Why is it that whenever I convert a bitmap, it takes a long time and then PixPro either tells me '**Allocated Memory Used!**' or the converted bitmap has a lot of polygons?

**A:** Check to see that your bitmap has been reduced down to a reasonable number of colors, such as 2 or 4. Then make sure your bitmap doesn't have a lot of dithering in it. A nicely shaded picture looks great as a picture, but PixPro tries to convert each and every different pixel to a polygon, and that takes time and uses memory. Try to include large solid colored areas in your bitmap, they tend to work best. Note that in most cases PixPro should have enough allocated memory to convert your bitmap. Also, the conversion time should be quick (1-2 minutes tops). If either of these do not seem to be true, you should check for other problems.

**Q:** What does '**Allocated memory used!**' mean? I have tons of ram!

**A:** '**Allocated memory used!**' means that PixPro has used up all the memory that it has allocated for itself when it is first started. PixPro normally allocates enough memory to accommodate 22,500 vertices. If you try to do something

that exceeds this limit, such as load a very large object, you will get this error. PixPro has the potential to load up to a maximum of 32,768 vertices and unlimited polygons. To allocate more than the default 22,500 vertices, press the **Default** button and then the **Memory** button. You only need to click on the UP arrow a couple of times to allocate more memory. You can go past 32,760, but it is not recommended to load objects that have more than 32,768 points since your objects will start to lose polygons after the maximum number of points has been reached. We recommend a maximum of 35,700 vertices allocated. It has been our experience that 99% of objects load in with this setting.

**Q:** Why does my converted bitmap look OK, but has a lot of points.

**A:** Did you use the **Smoothing** requester? The **Smoothing** requester is built especially for eliminating points and making your object look GREAT!

**Q:** When I load a PixPro converted object into LightWave, the front face appear divided. Why is this?

**A:** If your converted logo has divided faces on the front, it is most likely that the original polygon contained over 200 sides. LightWave will not accept polygons with more than 200 sides and so PixPro automatically divides such polygons into smaller loadable polygons when saving in certain formats such as LightWave. NOW, the most common reason people have these large polygons is because they were never smoothed. If you converted a bitmap and did not use the **Smoothing** requester to clean up the object, you will in many cases get very large polygons, the very kind LightWave cannot load.

**Q:** I tried loading in my DXF file but I get the error 'Unrecognized file format!'. What am I doing wrong?

**A:** In most cases, the DXF file that you are trying to load has been converted from a PC. A DXF file is a readable ASCII (text) file. If you are using a program such as Cross Dos to bring the file over to your Amiga, you need to use the Text Filter option. The filter will turn all the line feeds and carriage returns of the ASCII file into just carriage returns, which is what PixPro wants. If turning on the text filter in Cross Dos does not work, make sure your file is an ASCII (text) kind of DXF file instead of a Binary one. PixPro's DXF loader is not specific enough for it to matter which version of DXF file you have.

**Q:** Why when I convert my Imagine or Sculpt files to LightWave and then render them, there are a bunch of holes everywhere?

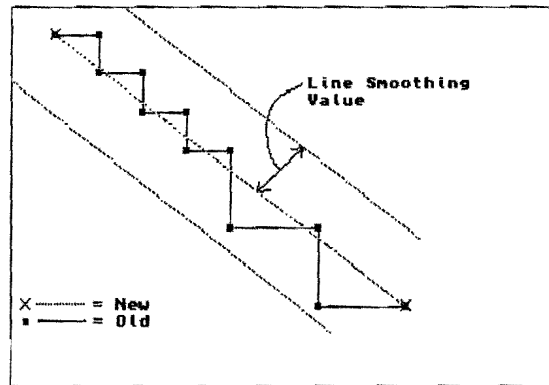
**A:** Imagine and Sculpt do not care whether their polygons are ordered clockwise or counter clockwise, but LightWave does. There are three ways of solving this problem. The most efficient is to use PixPro's **Polygon Flipping** function in the **Data** requester. This function will intelligently flip all your polygons so they will all render properly. The second and less efficient way is to simply hit the Double Sided button in LightWave's Surface requester but LightWave ends up rendering more polygons. The third and least efficient way (if for some reason the first two don't work) is to use PixPro's **Double Polygon** function. This function will create a polygon going the opposite way for every polygon in the object, thus twice as many polygons will be created.



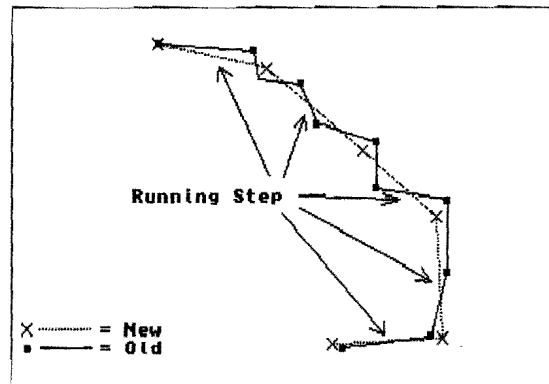
**Q:** PixPro is not loading my picture I saved out of my Video Toaster, what is the problem?

**A:** When you save a picture out of ToasterPaint to be used with PixPro, it needs to be saved as an RGB file and not as a frame. PixPro, however will not load 24 bit RGB files as they contain too many colors for a practical conversion. The RGB file saved from ToasterPaint must be converted to an IFF bitmap that has less then 32 colors. Programs such as ADPro and ImageFX will convert these files. Once you have converted the file, it is usually a good idea to edit the bitmap (picture) in a painting program such as Deluxe Paint. In Deluxe Paint you can eliminate dithered areas, lone pixels and extra jaggie areas. The idea is to make the bitmap look like super clean clip art. Then, load the bitmap into PixPro and go!

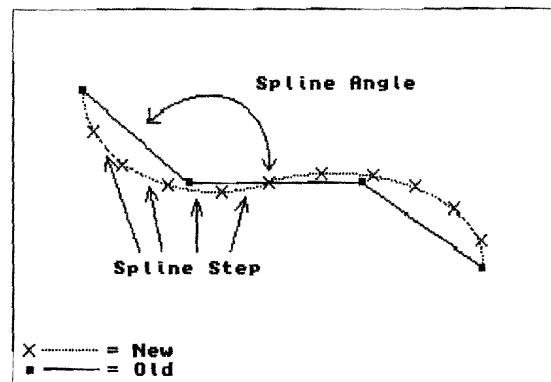
Picture 1 (See Section 8.1)



Picture 2 (See Section 8.1)



Picture 3 (See Section 8.1)



# ACKNOWLEDGMENTS

## **Programming:**

Scott Greyson Thede

## **Additional Programming:**

Brian Wagner, Michael Plitkins, Greg Gorby

## **Manual:**

John Gross

The programmer, Scott Greyson Thede (known lovingly as Pixel Boy) would like to thank all the wonderfully, amazingly talented people that have been involved in the making of this project:

.. Brian Wagner, John Gross, Kathy & Lee Stranahan, James Flinn, Michael Plitkins, Bob Anderson, Greg Gorby, Jim Plant, Allen Hastings, Stuart Ferguson, Yarg Ymmat, Victor Osaka, Leo Martin, Loa Thede, Steve Thede, Sharon Thede, Shane Thede (Go Marines!). NewTek: Tim, Paul, Kiki, Mark, Daniel, Christina

Special greetings to my techno buddies: Eric Donaldson, Walter Reed, Dan Stephenson.